Clouds and the Earth's Radiant Energy System (CERES)

Data Management System

Software Design Document

Gridding

(Subsystems 6.0 & 9.0)

Architectural Draft

Troy Anselmo[2], Kay Costulis[1], Maria Mitchum[1], Nichele McKoy[2], Joe Stassi[2]

[1]NASA Langley Research Center
Mail Stop 423
Hampton, VA 23681-0001

[2]Science Applications International Corporation (SAIC)
One Enterprise Parkway, Suite 300
Hampton, Virginia 23666

Atmospheric Sciences Division
NASA Langley Research Center
Hampton, Virginia 23681-0001

Release 1.0
May1996

Preface

The Clouds and the Earth's Radiant Energy System (CERES) Data Management System supports the data processing needs of the CERES science research to increase understanding of the Earth's climate and radiant environment. The CERES Data Management Team works with the CERES Science Team to develop the software necessary to support the science algorithms. This software, being developed to operate at the Langley Distributed Active Archive Center (DAAC), produces an extensive set of science data products.

The Data Management System consists of 12 subsystems; each subsystem represents a stand-alone executable program. Each subsystem executes when all of its required input data sets are available and produces one or more archival science products.

The documentation for each subsystem describes the software design at various stages of the development process and includes items such as Software Requirements Documents, Data Products Catalogs, Software Design Documents, Software Test Plans, and User's Guides.

This version of the Software Design Document records the architectural design of each Subsystem for Release 1 code development and testing of the CERES science algorithms. This is a PRELIMINARY document, intended for internal distribution only. Its primary purpose is to record what was done to accomplish Release 1 development and to be used as a reference for Release 2 development.

Acknowledgments are made to David F. Young, Analytical Services & Materials, Inc., and G. Louis Smith, NASA Langley Research Center, for their science support.

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

List of Figures

1.0 Introduction

1.1 Document Overview

The purpose of this document is to explain the design specifications which were used to guide the development of the Grid Single Satellite Radiative Fluxes and Clouds (Subsystems 6.0 and 6.1) and Grid TOA and Surface Fluxes (Subsystems 9.0 and 9.1). The intended audience for this document includes the Subsystem design team, programmers, testers, follow-on subsystem design teams, and science reviewers.

The Design Document contains the following information:

- Document Overview
- CERES System Overview
- A brief overview of the Subsystems' purpose and functionality
- A description of the key concepts embodied in the subsystem design
- Descriptions of constraints placed on the design and implementation of the software
- The Architectural Design consisting of class diagrams, scenario diagrams, and text which describes the components which comprise the system as well as the interaction of these components
- References
- Appendix A - Abbreviations and Acronyms

1.2 CERES System Overview

The Clouds and the Earth's Radiant Energy System (CERES) is a key component of the Earth Observing System (EOS). The CERES instruments are improved models of the Earth Radiation Budget Experiment (ERBE) scanner instruments, which operated from 1984 through 1990 on the National Aeronautics and Space Administration's (NASA) Earth Radiation Budget Satellite (ERBS) and on the National Oceanic and Atmospheric Administration's (NOAA) operational weather satellites NOAA-9 and NOAA-10. The strategy of flying instruments on Sun-synchronous, polar orbiting satellites, such as NOAA-9 and NOAA-10, simultaneously with instruments on satellites that have precessing orbits in lower inclinations, such as ERBS, was successfully developed in ERBE to reduce time sampling errors. CERES will continue that strategy by flying instruments on the polar orbiting EOS platforms simultaneously with an instrument on the Tropical Rainfall Measuring Mission (TRMM) spacecraft, which has an orbital inclination of 35 degrees. In addition, to reduce the uncertainty in data interpretation and to improve the consistency between the cloud parameters and the radiation fields, CERES will include cloud imager data and other atmospheric parameters. The first CERES instrument is scheduled to be launched on the TRMM spacecraft in 1997. Additional CERES instruments will fly on the EOS Morning Crossing Mission (EOS-AM) platforms, the first of which is scheduled for launch in 1998, and on the EOS Afternoon Crossing Mission (EOS-PM) platforms, the first of which is scheduled for launch in 2000.

1.3 Subsystem Overview

The Grid Single Satellite Radiative Fluxes and Clouds and Grid TOA and Surface Fluxes Subsystems provide the transformation from instantaneous, instrument-referenced data to spatially-averaged Earth-referenced data. These Subsystems perform two major functions: gridding and spatial averaging. The gridding function assigns CERES footprints to the appropriate regional hourbox. The spatial averaging function computes spatial averages of the various radiative flux parameters and cloud properties over each regional hourbox. After passing through this Subsystem, the CERES data lose their traceability to specific CERES measurements.

1.3.1 Subsystem 6.0 and 6.1

The primary input to Subsystem 6.0 is the Single Satellite CERES Footprint Radiative Fluxes and Clouds (CRS) data products, which is an hourly collection of CERES footprints containing longwave and shortwave radiative fluxes for the surface, internal atmosphere, and Top-of-the-Atmosphere (TOA). The primary output of the Subsystem is the Hourly Gridded Single Satellite Fluxes and Clouds (FSW) product which contains the TOA, atmospheric, and surface fluxes spatially averaged for each regional hourbox observed during the hour.

1.3.2 Subsystem 9.0 and 9.1

The primary input to Subsystem 9.0 is the Single Satellite CERES Footprint TOA and Surface Fluxes (SSF) data product, which is an hourly collection of CERES footprints containing radiative fluxes and cloud information. The primary output of this Subsystem is the Hourly Gridded Single Satellite TOA and Surface Fluxes (SFC) data product, which contains radiative fluxes and cloud properties spatially averaged by regional hourbox.

Details of each of the four data products (CRS, FSW, SSF, and SFC) discussed above can be found in the CERES Data Management System Data Products Catalog (Reference 1).

1.4 Key Concepts

There are several key operations which form the core of the gridding subsystems and are important to understanding the code which implements these subsystems. These key operations include:

- processing the data in chunks
- reconstructing scan lines from the serial stream of footprints presented by the SSF and CRS input data sets
- sharpening footprints
- assigning footprints to the correct regional hourbox

- averaging the footprints in each regional hourbox
- postprocessing to sort the hourly FSW / SFC data into the proper order for zonal files

1.4.1 Processing Data in Chunks

When the architectural design was developed, the design team did not know how much core memory was going to be available on the production computers. Therefore, to insure that the Subsystems would be able to accommodate the available resources, the decision was made to break the input data sets into a series of "chunks" where each chunk represents a certain number of scan lines worth of data. The Subsystem reads in enough data from the input file to fill up a chunk and then processes the footprints in that chunk. This process is repeated until the entire input data set is read and processed. Although this provides a degree of flexibility when running on different machines such as those found in the SCF and at the DAAC, it introduces boundary conditions at the interfaces between chunks which the Subsystem must accommodate. The problems introduced by these boundary conditions and how they are handled is addressed in Section 1.4.2 and Section 1.4.5.

1.4.2 Reconstruct Scan Lines from a Serial Stream of Footprints

The algorithms used to sharpen and average footprints were provided by the CERES Science Team. These algorithms depend on the positions of footprints relative to one another in order to work correctly. This requires the Subsystem to organize the footprints into scan lines that represent how the data was originally obtained by the instrument.

1.4.3 Sharpen Footprints

In an effort to improve the accuracy of the data, the Gridding Subsystems attempt to sharpen the flux parameters of the footprints. The algorithm used to sharpen the flux parameters is provided by the CERES Science Team. The basic concept of sharpening footprints is to perform a weighted sum of the footprint to be sharpened and its surrounding neighbors for each flux parameter to enhance their value. Below is a detailed explanation of the sharpening concept.

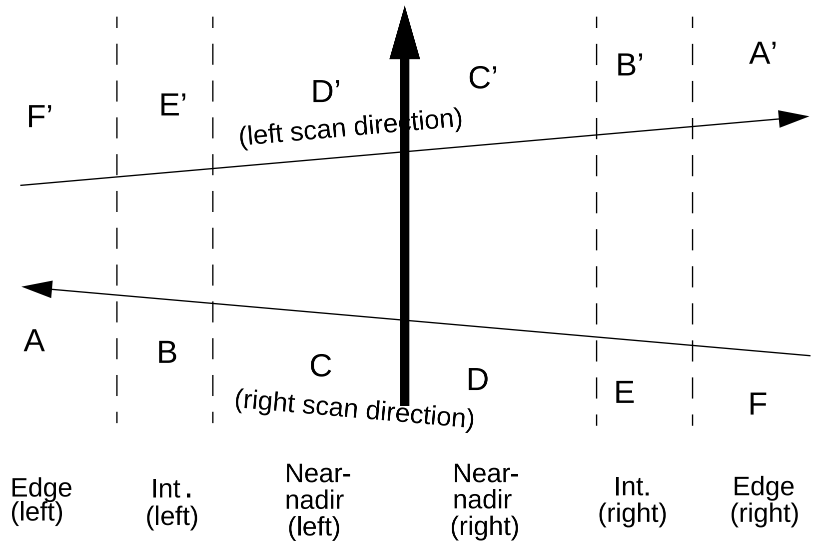
Time Interpolation and Spatial Averaging (TISA) Gridding TOA Flux Sharpening

1) Sharpening Equation: Given a 7x7 matrix of TOA flux values, the center value is sharpened with the following equation.

$$sharpenedflux\,(4, 4) = \sum_{i=1}^{7} \sum_{j=1}^{7} weight\,(i, j) \cdot flux\,(i, j)$$

2) There are six weight zones in each scan (where "scan" is defined as a single sweep across the Earth): Near-nadir (left and right), Intermediate (left and right), and Edge (left and right). The weight zone breakpoints are given in terms of the instrument cone angle. The specific breakpoints to be used for CERES processing have yet to be determined.

.



3) The set of weights used in the sharpening algorithm will be determined by the weight zone of the footprint being sharpened and the scan direction, right or left, at the time of the footprint measurement. The right and left scan direction weights are mirror images of each other; i.e., for weight sets A and A', A(i,j)=A'(i,8-j) for i=i,7 and j=1,7.

4) Three flux values at the extremities will **not** be sharpened. These include the three right-most and the three left-most footprints of the scan, the footprints in the last three scans prior to a data gap, and the footprints in the first three scans following a data gap.

5) If single footprints are missing, they will be replaced with interpolated values calculated using corresponding footprints in the next and previous scans. (Question: Do we want to interpolate two right direction footprints to get a left direction footprint?)

6) Each satellite (instrument?) will have its own sets of weights.

7) For each region, a quality flag will be set (for each channel?) with one of the following values:

0--none of the footprint values were sharpened prior to averaging
1--some of the footprint values were sharpened prior to averaging
2--all of the footprint values were sharpened prior to averaging
Or another possibility is that we output a percentage of the flux values sharpened in a region, where 0=none and 100=all.

1.4.4 Assign Footprints to the Correct Regional Hourbox

The regional hourbox that a footprint is assigned to is based on the region number and hour number associated with the given footprint. The region number assigned to a footprint is determined by the colatitude and longitude of the footprint and the CERES Reference Grid (Reference 2). In Subsystem 6.0, the Greenwich meridian time (GMT) hour is used for the hour number of the footprint. In Subsystem 9.0, the local hour of the center of the regional hour box is used to assign the footprint to the appropriate hourbox. An algorithm provided by the CERES Science Team is used to calculate the local hour number.

1.4.5 Average Footprints in Each Regional Hourbox

The Gridding Subsystems read in footprints from the CRS / SSF input files. These footprints are binned into their appropriate regional hourbox to be averaged (Section 1.4.4). Before an hourbox can be averaged, it must deal with chunk boundaries and file boundaries. The chunk boundaries are handled by only averaging the hourboxes which have been passed to the hourbox averaging routine twice. The file boundary problem only occurs in Subsystem 9.0 due to processing at local time. In Subsystem 9.0, it is necessary to process one chunk into the next consecutive SSF input file in order to average any remaining hourboxes. In Figure 1-1, it is necessary to read one chunk into FILE #2 to average region number 24542; it is possible that FILE #2 contains footprints which belong to hourboxes in FILE #1. The hourboxes will be averaged by either an arithmetic mean or a cubic spline interpolation. The cubic spline interpolation will only be done if all the footprints were sharpened (Section 1.4.3), otherwise, an arithmetic mean will be calculated for the hourbox.

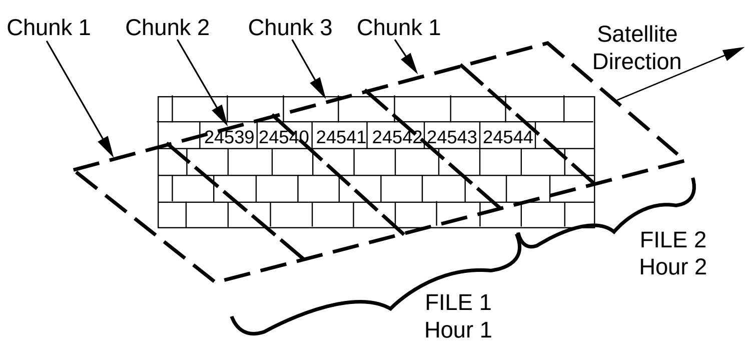


Figure 1-1. File Boundaries

1.4.6 Post Process to Get Zonal Files Sorted by Region and Then by Hour

The CRS and SSF input products to the Gridding Subsystems are hourly products; and the main processor, Subsystem 6.0 and Subsystem 9.0, handles these files on an hourly bases. The final product of the Gridding Subsystems is a monthly zonal product which is sorted by region and then by hour. Therefore, it is necessary to postprocess the data for two reasons: to create a monthly product and to properly order the data. The postprocessor takes all the intermediate hourly products for a given month and creates a monthly data product which is organized in zonal files that are sorted by region and then by hour.

1.5 Implementation Constraints

Metadata and Error Handling. Metadata and error handling will conform to the EOSDIS operating environment requirements, which are currently being defined. Metadata are discussed in the External Interface Requirements section of this document. Details regarding error handling have been deferred until more information on the operating environment is available. We assume that there will be a system-level set of error messages that will be invoked during processing. Fatal errors will result in a “graceful” halt to processing and diagnostic messages. Nonfatal errors will result in diagnostic messages to the operator/analyst and to the Quality Control (QC) reports.

Compiler Constraint. The maximum number of files allowed open by most FORTRAN 90 compilers is 99, which caused an implementation constraint in the post_processor. The design team originally intended to open all 744 hour files for a month, and sort the hour files into zonal files; however, as a result of only being able to have 99 files open at a time, it was necessary to first reorganize the hour files into day files and then sort the day files into zonal files. This created an additional step in the design of the post_processor thus increasing the number of IO calls which increases the execution time.

1.6 Design Approach

The TISA Gridding software is designed to create a single executable that will process both Subsystem 6.0 and Subsystem 9.0. The main difference between the Subsystems is that Subsystem 6 is processed using GMT time of the footprints and Subsystem 9 is processed using local time of the footprints. Processing of the data is the same except for binning of the footprints into the appropriate hourbox (see Section 1.4.4). The design team decided that using a single set of code for Subsystems 6 and 9 allows them to take advantage of the duplication of code between the Subsystems and will decrease the amount of time and liability required for maintenance.

2.0 Architectural Design

This section contains the context and scenario diagrams which provide a pictorial description of the TISA Gridding Subsystems. Following each context diagram is a brief overview describing the diagram, and following each scenario diagram is a description of the steps depicted in the diagram.

2.1 TISA Gridding Main Processor

The following context diagram depicts the USE relationship of the FORTRAN 90 modules and the main driver of the TISA Gridding Main Processor.

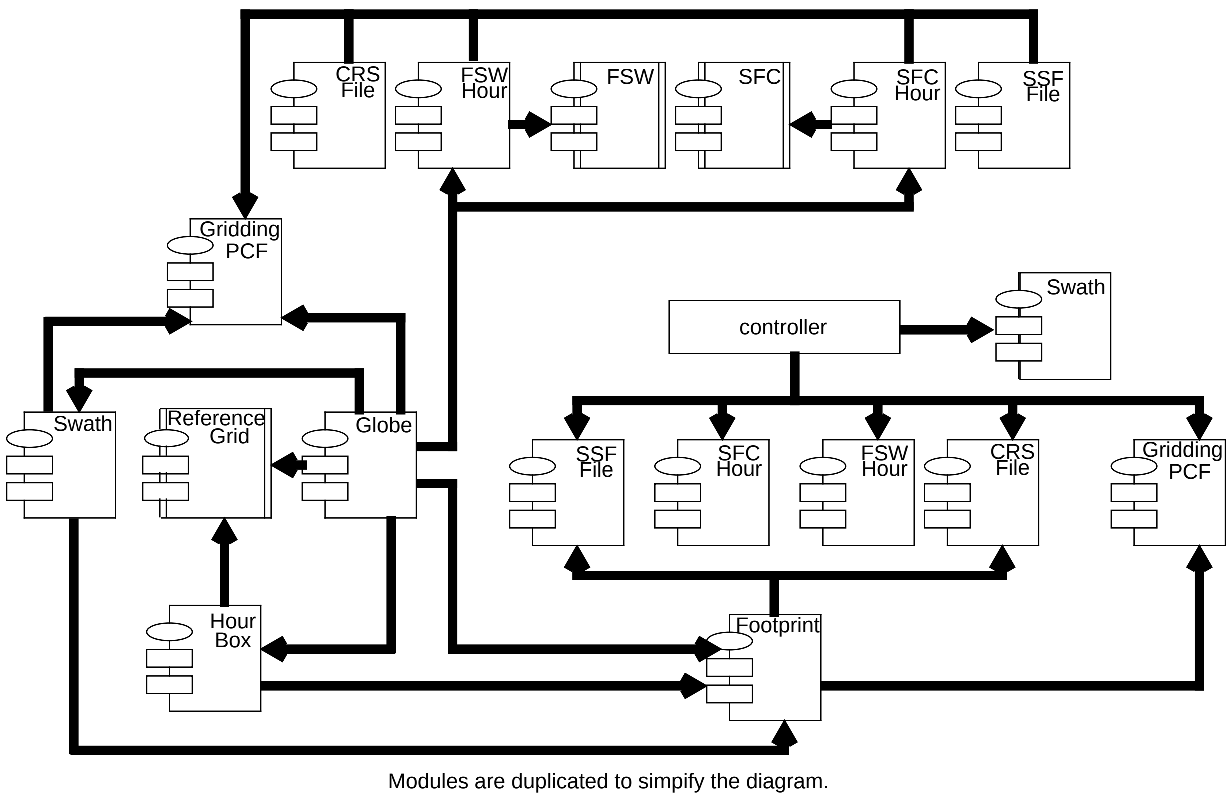


Figure 2-1. TISA Gridding Main Processor Context Diagram

A brief overview of the main driver and the modules in Figure 2-1 is given below.

2.1.1 controller

The contoller is the driver of the TISA Gridding Main Processor.

2.1.2 CRS_File

The CRS_File module acts as a wrapper to the CRS_IO module provided by Subsystem 5.0. It validates the header of the CRS data file, reads the data records from the CRS data files, and retrieves the data from the data record that is needed by Subsystem 6.0.

2.1.3 SSF_File

The SSF_File module acts as a wrapper to the SSF_TYPDEF module provided by Subsystem 4.6. It validates the header of the SSF data file, reads the data records from the SSF data files, and retrieves the data from the data record that is needed by Subsystem 9.0.

2.1.4 FSW_Hour

This FSW_Hour module gets the data needed to create the output product for Subsystem 6.0.

2.1.5 FSW

The FSW module provides the I/O interface to the FSW data product.

2.1.6 SFC_Hour

The SFC_Hour module gets the data needed to create the output product for Subsystem 9.0.

2.1.7 SFC

The SFC module provides the I/O interface to the SFC data product.

2.1.8 Gridding_PCF

The Gridding_PCF module provides a wrapper to the PCF module found in CERESlib. It provides a specific interface to the TISA Gridding Subsystems that would not be achieved by the PCF module.

2.1.9 Footprint

The footprint module provides two abstract data types called footprint_item_type and avg_footprint_type and the routines that can perform operations on these types.

2.1.10 Hourbox

The Hourbox module provides an abstract data type called hourbox_type and the routines that can perform operations on this type. It a private type which hides the implementation details for an hourbox. The hourbox_type is implemented as a structure which contains all the data that constitutes an hourbox such as the region number, the hour number, a variable size list of footprints which fall in the hourbox, and an average footprint for the hourbox.

2.1.11 Globe

The Globe module provides an abstract state machine called globe which is essentially a collection of hourboxes.

2.1.12 Swath

The Swath module contains routines that recreate the instrument scan line geometry from a serial stream of footprints provided in the SSF or CRS input file. The module creates a 2D array which represents the chunk of footprints. The module works with a chunk or subset of the total number of footprints found in the input file and can iterate through these chunks until the entire file has been processed. The module detects and accommodates data gaps due to missing data as well as instrument resets.

2.1.13 Reference Grid

The Reference Grid module provides information about the CERES Reference Grid, and it is needed to determine the region number to assign to the footprint.

Figure 2-2 depicts the modules that define the underlying abstract data types used to compose a footprint.

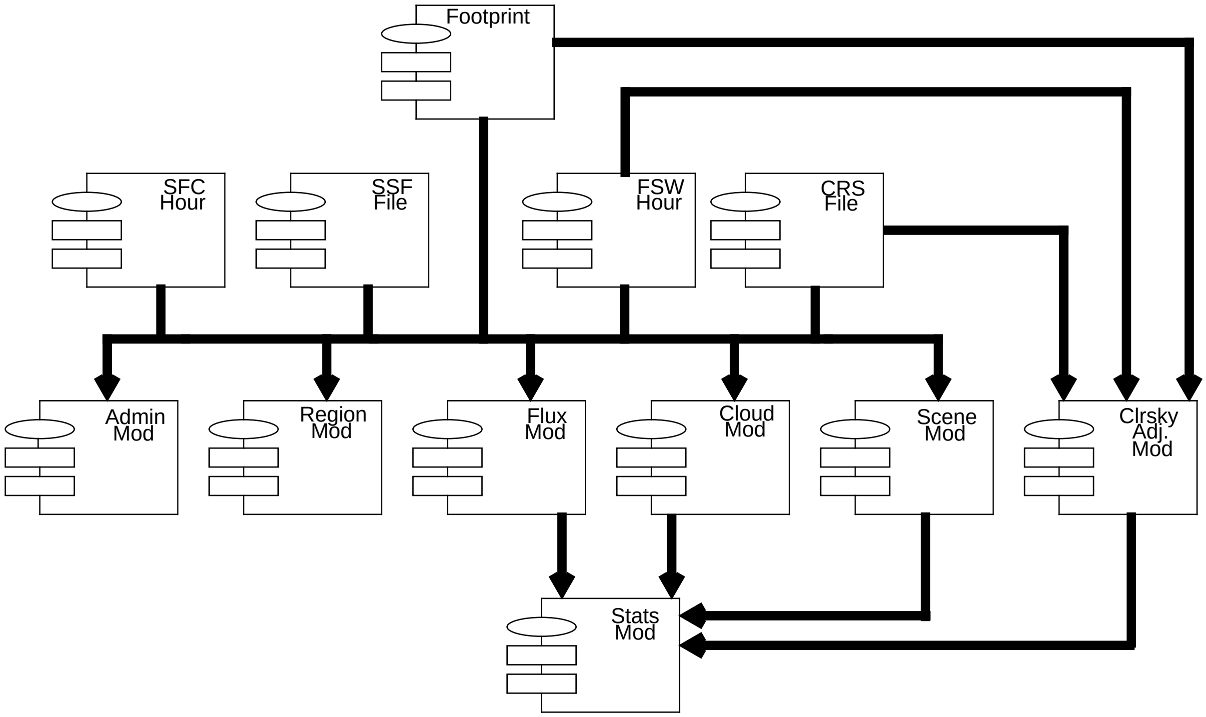


Figure 2-2. TISA Gridding Main Processing Context Diagram

A brief overview of the modules in Figure 2-2 is given below.

2.1.14 Footprint

See Section 2.1.9.

2.1.15 CRS_File

See Section 2.1.2.

2.1.16 FSW_Hour

See Section 2.1.4.

2.1.17 SSF_File

See Section 2.1.3.

2.1.18 SFC_Hour

See Section 2.1.6.

2.1.19 Admin_Mod

The Admin_Mod module provides two abstract data types called admin_type and avg_admin_type and the routines that can perform operations on these types.

2.1.20 Region_Mod

The Region_Mod modules provides two abstract data types called region_type and avg_region_type and the routines that can perform operations on these types.

2.1.21 Flux_Mod

The Flux_Mod module provides several abstract data types for the flux parameters and an abstract data type called avg_flux_type. This module also provides the abstract data types called srf_only_type and the avg_srf_only type which are used for the surface only parameters.

2.1.22 Cloud_Mod

The Cloud_Mod module provides two abstract data types called cloud_type and avg_cloud_type and the routines that can perform operations on these types.

2.1.23 Scene_Mod

The Scene_Mod module provides two abstract data types called scene_type and avg_scene_type and the routines that can perform operations on these types.

2.1.24 Clrsky_Adj._Mod

The Clrsky_Adj._Mod module provides two abstract data types called clrsky_adj_type and avg_clrsky_type and the routines that can perform operations on these types.

2.1.25 Stats_Mod

The Stats_Mod module provides two abstract data type called stats and stats2 and the routines that can perform operations on these types.

2.2 TISA Gridding Post_Processor

The following context diagram depicts the USE relationship of the FORTRAN 90 modules and the main driver of the TISA Gridding Post_Processor.

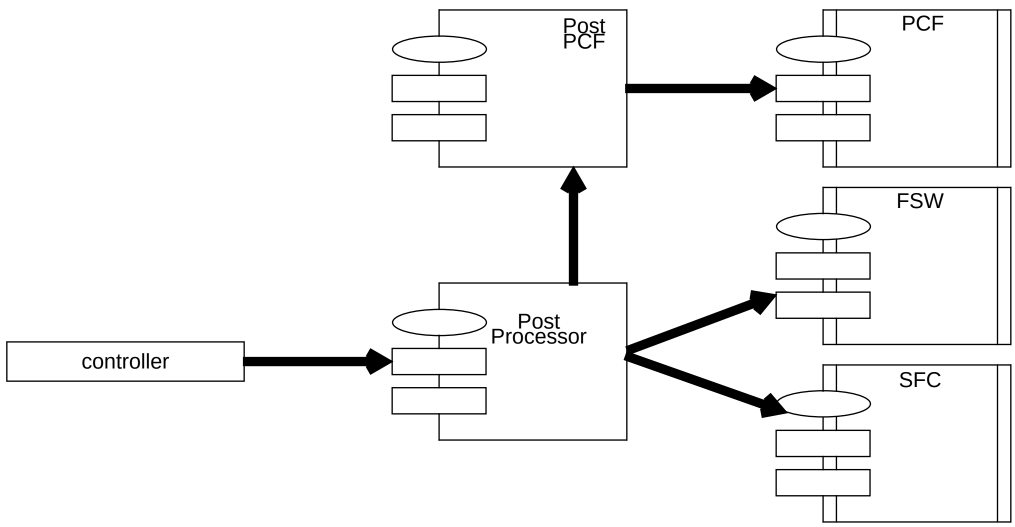


Figure 2-3. TISA Gridding Post-Processor Context Diagram

A brief overview of the main driver and the modules in Figure 2-3 is given below.

2.2.1 controller

The controller is the driver of the TISA Gridding Postprocessor.

2.2.2 Post_Processor

The Post_Processor module creates a monthly FSW product for Subsystem 6.0 or a monthly SFC product for Subsystem 9.0 that is organized in zonal files that are sorted by region and then hour.

2.2.3 Post_PCF

The Post_PCF module provides a wrapper to the PCF module found in CERESlib. It provides a specific interface to the TISA Gridding Subsystems that would not be achieved by the PCF module.

2.3 TISA Gridding Initialization Scenario

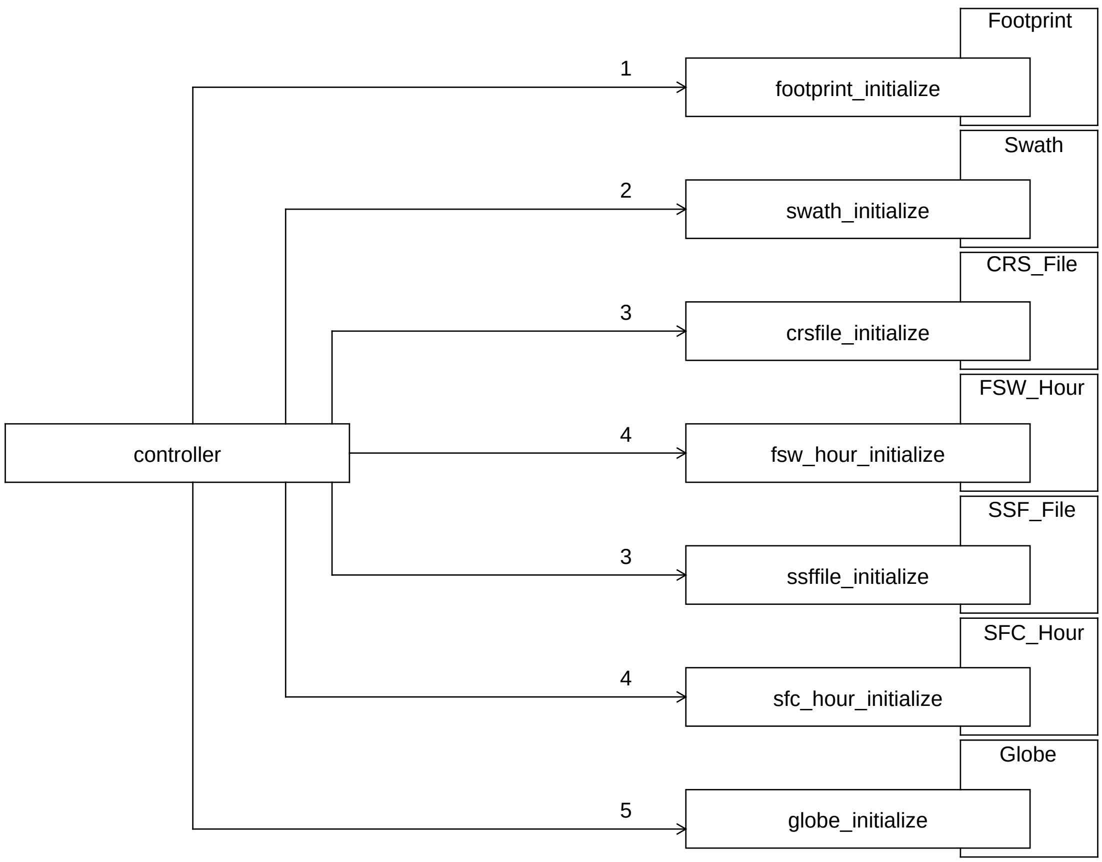


Figure 2-4. TISA Gridding Initialization Scenario Diagram

Below is a step-by-step description of the processing steps depicted in Figure 2-4.

1) **footprint_initialize:** The footprint module initializes itself by determining which subsystem it is processing.

2) **swath_initialize:** The swath module initializes itself by determining the chunk size that will be processed and allocating memory to hold the data chunk.

3) This step depends on the subsystem that is being processed. If Subsystem 6.0 is being processed, then crsfile_initialize is called. If Subsystem 9.0 is being processed, then ssffile_initialize is called.

   crsfile_initialize: The CRS File module opens the CRS data file that will be used for input and validates the header of the CRS File. It also determines, from the Gridding PCF module, how many footprints to read in at a time.

   ssffile_initialize: The SSF File module opens the SSF data file that will be used for input and validates the header of the SSF File. It also determines, from the Gridding PCF module, how many footprints to read in at a time.

4) This step depends on the subsystem that is being processed. If Subsystem 6.0 is being processed, then fsw_hour_initialize is called. If Subsystem 9.0 is being processed, then sfc_hour_initialize is called.

   fsw_hour_initialize: The FSW Hour module opens the hourly FSW data file for output.
   sfc_hour_initialize: The SFC Hour module opens the hourly SFC data file for output.

5) **globe_initialize:** The Globe module initializes the globe by creating a new list of hourboxes and gathering time information from the Gridding PCF module that will be needed to determine which regional hourbox to bin to a footprint.

2.4 TISA Gridding Main Processing Scenario

Figure 2-5 is the scenario diagram describing the processing which occurs during the TISA Gridding Main Processor.

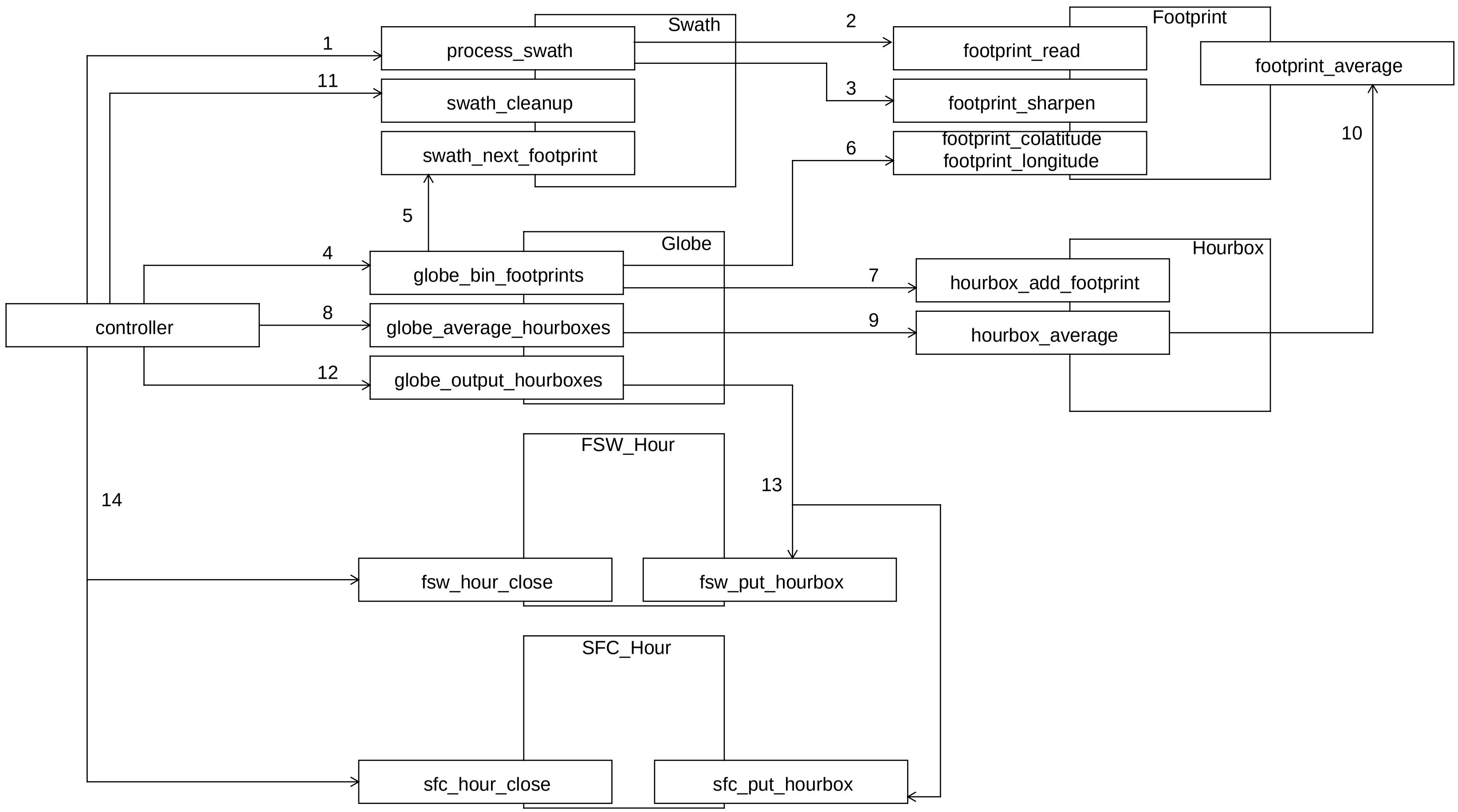


Figure 2-5. TISA Gridding Main Processing Scenario Diagram

Below is a step-by-step description of the processing steps depicted in Figure 2-5.

1) **process_swath:** The Swath module reads in the next chunk of data from the footprint_read routine in the footprint module and then sharpens the footprints by passing each footprint and its neighbors to the footprint sharpen routine.

2) **footprint_read:** The Footprint module reads the next footprint from the CRS_File module for Subsystem 6.0 or the SSF File module for Subsystem 9.0.

3) **footprint_sharpen:** The Footprint module returns a sharpened footprint given a 2-dimensional array which consist of the footprint to be sharpened and its surrounding neighbors (see Section 1.4.3).

4) **globe_bin_footprint:** The Globe module bins the footprints returned by swath next footprint into its appropriate regional hourbox. The region number of the footprint is returned from Reference Grid module given the footprint_colatitude and the footprint_longitude. The hour number is based on the GMT hour of the CRS input file for Subsystem 6.0, and the calculated local hour which is returned from the convert_GMT_to_Local routine for Subsystem 9.0 (see Section 1.4.4).

5) **swath_next_footprint:** The Swath module returns the next footprint in its current chunk of data.

6) **footprint_colatitude:** The footprint module returns the colatitude of a given footprint.
   footprint_longitude: The footprint module returns the longitude of a given footprint.

7) **hourbox_add_footprint:** The Hourbox module adds the given footprint to the given hourbox.

8) **globe_average_hourboxes:** The Globe module traverses through its collection of hourboxes and obtains an average for each hourbox.

9) **hourbox_average:** The Hourbox module gathers all the footprints associated with the given region and calls the footprint_average routine to get an average hourbox.

10) **footprint_average:** The Footprint module takes in an array of footprints and calculates an averaged value for each parameter in the footprint.

11) **swath_cleanup:** The Swath module cleans up the swath by freeing all dynamically allocated memory that was used to process the swath.

12) **globe_output_hourboxes:** The Globe module traverse through its collection of hourboxes and outputs each hourbox to its appropriate output product through the means of FSW Hour for Subsystem 6.0 and SFC Hour for Subsystem 9.0.

13) This step depends on the subsystem that is being processed. If Subsystem 6.0 is being processed, then fsw_put_hourbox is called. If Subsystem 9.0 is being processed, then sfc_put_hourbox is called.
fsw_put_hourbox: The FSW_Hour module writes the given regional hourbox to an hourly FSW data file using the FSW module IO calls.
sfc_put_hourbox: The SFC_Hour module writes the given regional hourbox to an hourly SFC data file using the SFC module IO calls.

14) This step depends on the subsystem that is being processed. If Subsystem 6.0 is being processed, then fsw_hour_close is called. If Subsystem 9.0 is being processed, then sfc_hour_close is called.
fsw_hour_close: The FSW_Hour module writes a header to and closes the hourly FSW data file.
sfc_hour_close: The SFC_Hour module writes a header to and closes the hourly SFC data file.

2.5 TISA Gridding Post Processor

Figure 2-6 depicts the scenario diagram describing the processing which occurs during the TISA Gridding Post_Processor.

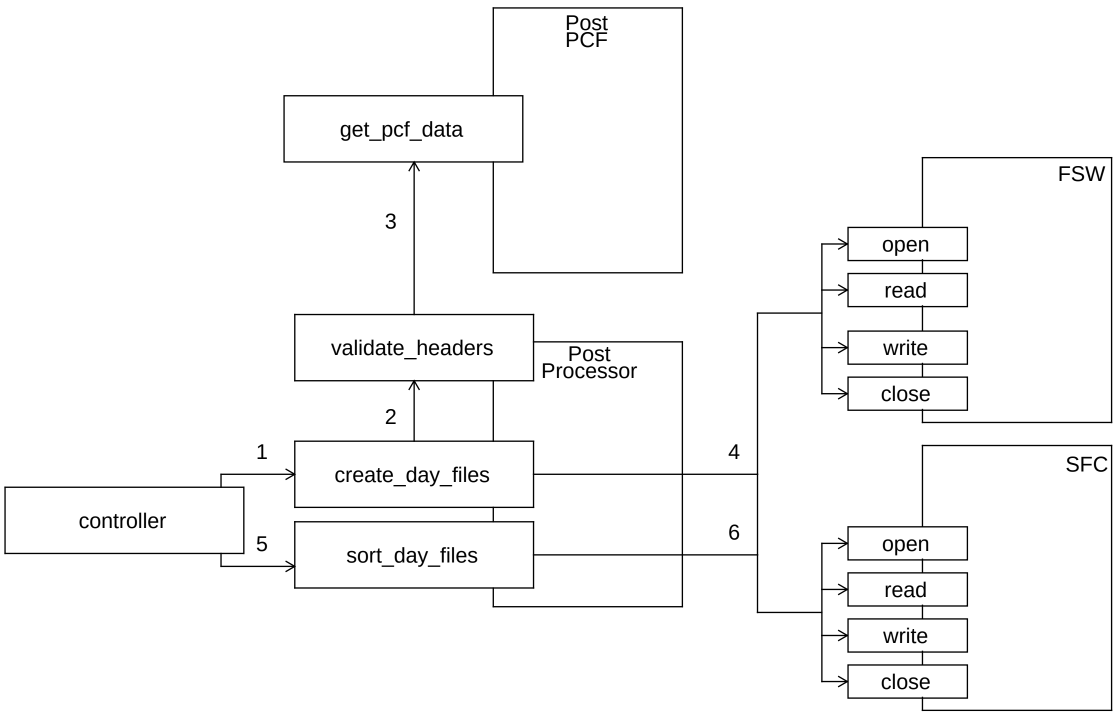


Figure 2-6. TISA Gridding Postprocessing Scenario Diagram

Below is a step-by-step description of the processing steps depicted in Figure 2-6.

1) **create_day_files:** The Post_Processor module reorganizes the hourly FSW / SFC data files into daily FSW / SFC data files. The headers from all of the hourly files are validated against information retrieved from the Post_PCF module. All of the day files (1 - 31) for a given month are opened. Then the hourly files are opened one at a time, and the hourbox data is transferred from the hourly file to the appropriate day file. As the hourboxes are read, an index of the hourboxes is created. Once all the hourly files have been read and written to the appropriate day file, the index is sorted by region and then hour within the region.

2) **validate_headers:** The headers of each hourly FSW / SFC data file is validated against information retrieved from the Post_PCF module.

3) **get_pcf_data:** The Post_PCF module returns the information retrieved from the Process Control file.

4) This step depends on the subsystem that is being processed. If Subsystem 6.0 is being processed, then FSW IO routines are called. If Subsystem 9.0 is being

processed, then SFC IO routines are called.
FSW IO routines: The FSW IO calls handle the file I/O operations that are performed to create the day files.
SFC IO routines: The SFC IO calls handle the file I/O operations that are performed to create the day files.

5) **sort_day_files:** The Post_Processor module creates a monthly data product which is organized in zonal files that are sorted by region and then hour. The zonal files are created from the day files using the sorted index that was created in Step 1.

6) This step depends on the subsystem that is being processed. If Subsystem 6.0 is being processed, then FSW IO routines are called. If Subsystem 9.0 is being processed, then SFC IO routines are called.
FSW IO calls: The FSW IO calls handle the file I/O operations that are performed to create the zonal files.
SFC IO calls: The SFC IO calls handle the file I/O operations that are performed to create the zonal files.

3.0 Modules

This section contains the detailed design diagrams for each module in the TISA Gridding Main and Post Processor.

3.1 CRS_File

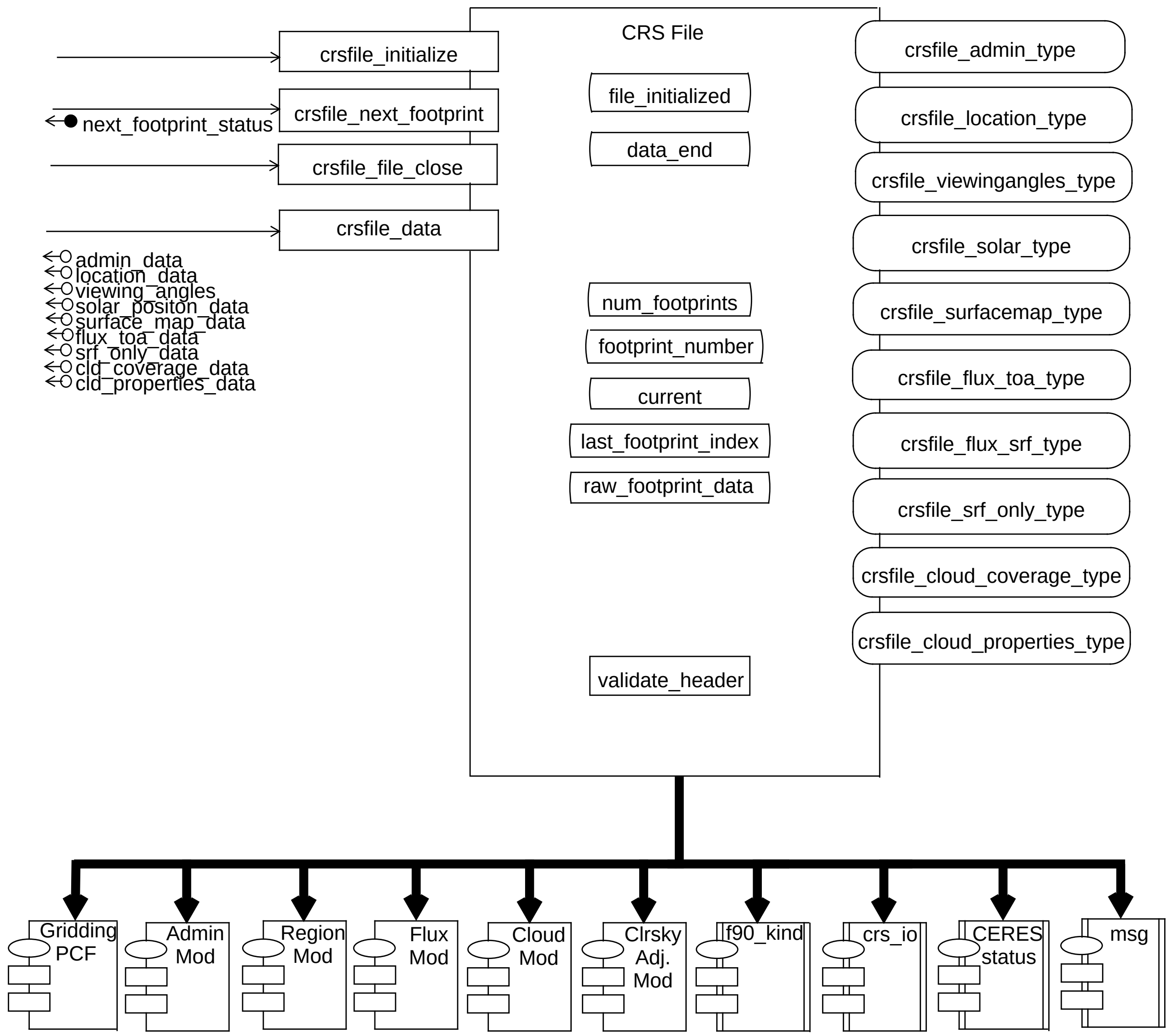


Figure 3-1. CRS_File Context Diagram

3.2 SSF_File

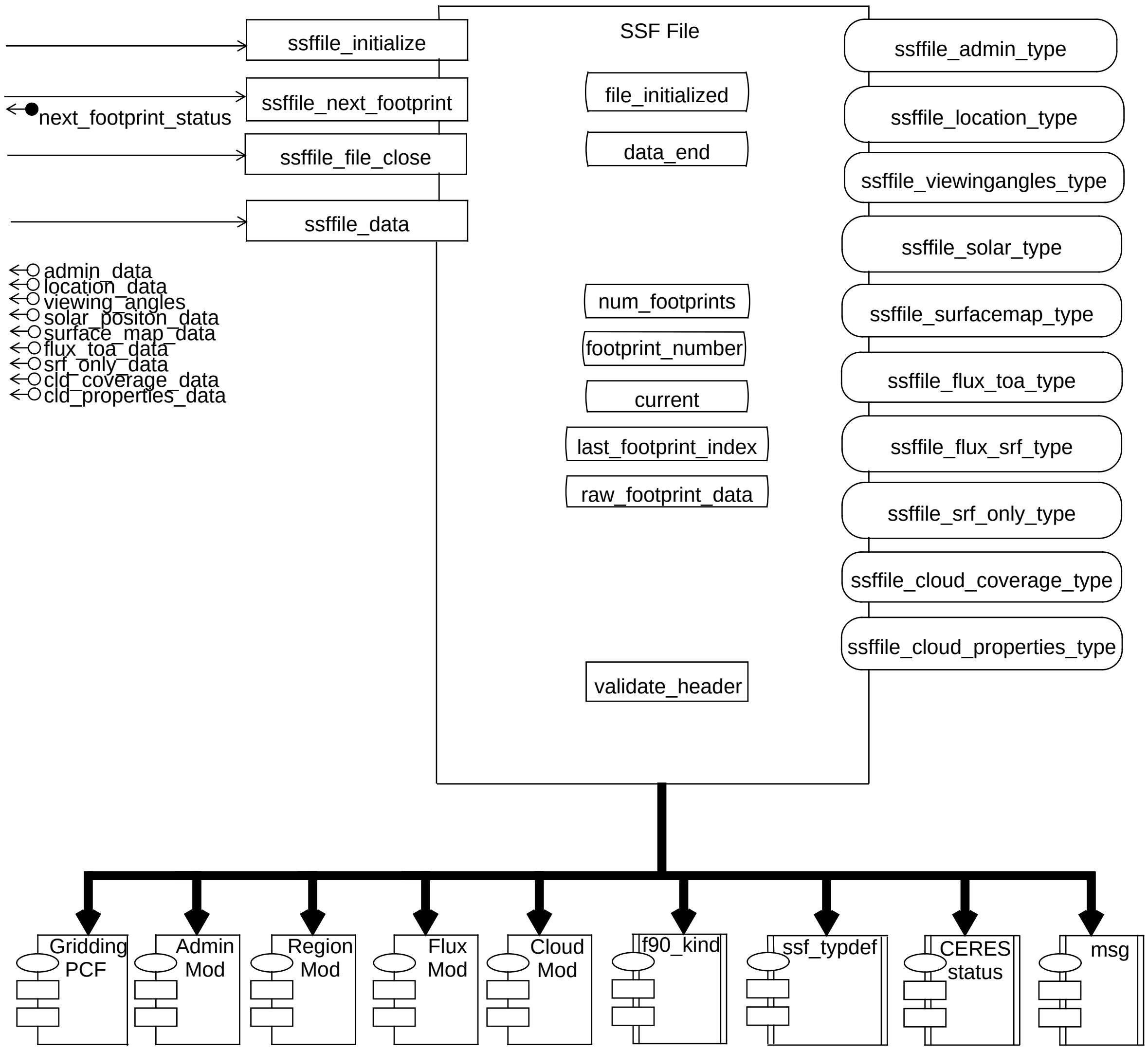


Figure 3-2. SSF_File Context Diagram

3.3 FSW_Hour

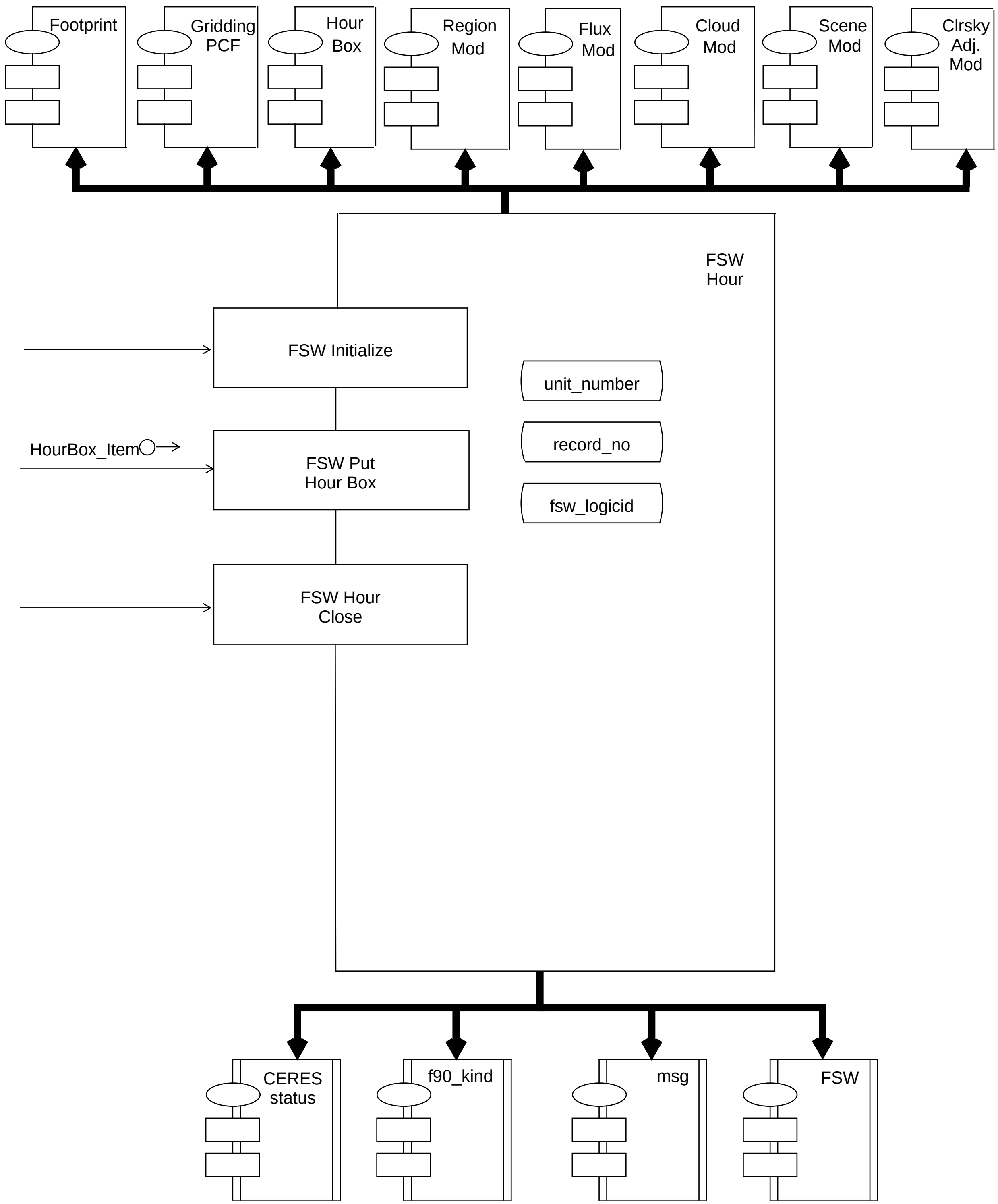


Figure 3-3. FSW_Hour Context Diagram

3.4 SCF_Hour

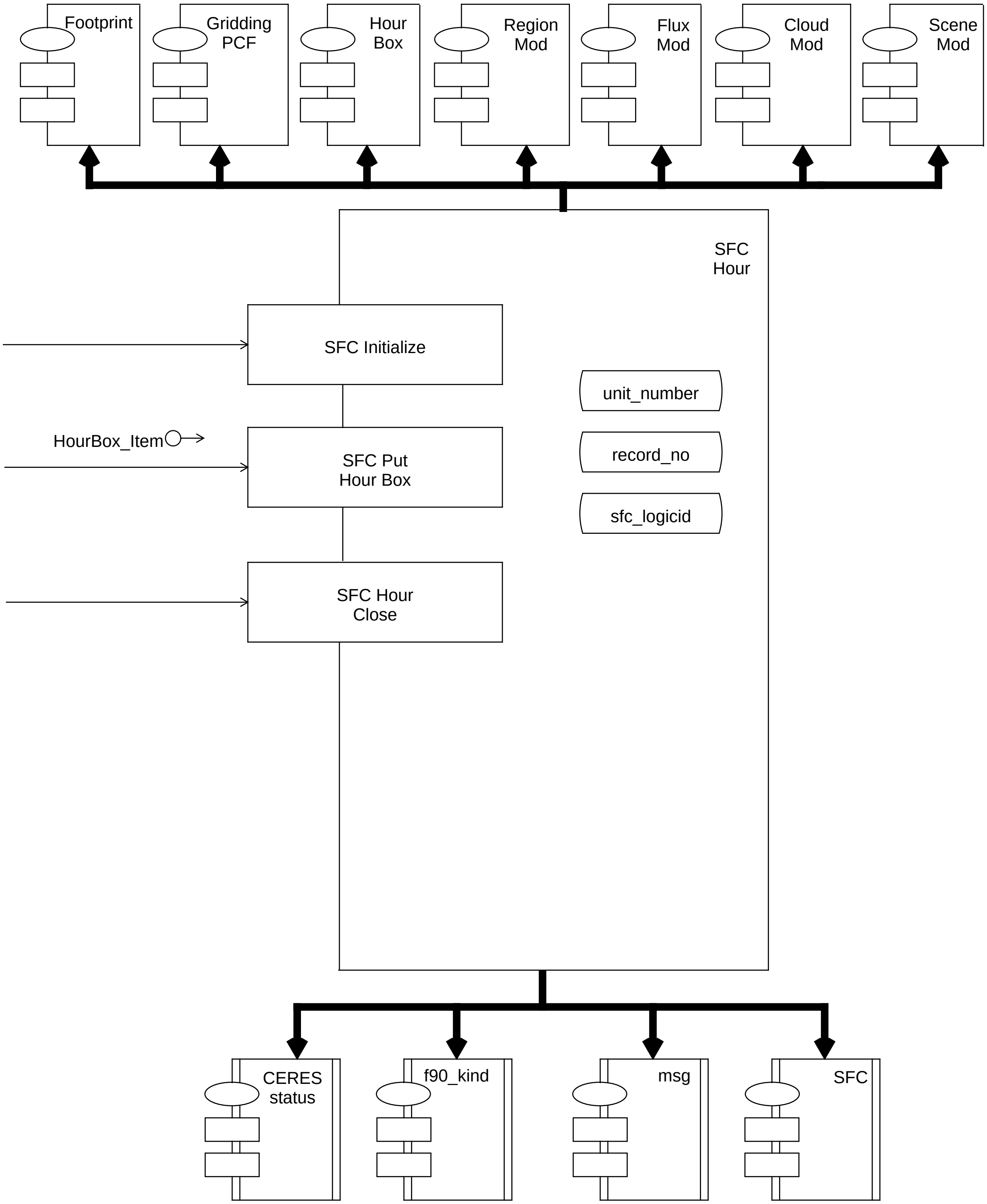


Figure 3-4. SFC_Hour Context Diagram

3.5 Gridding_PCF

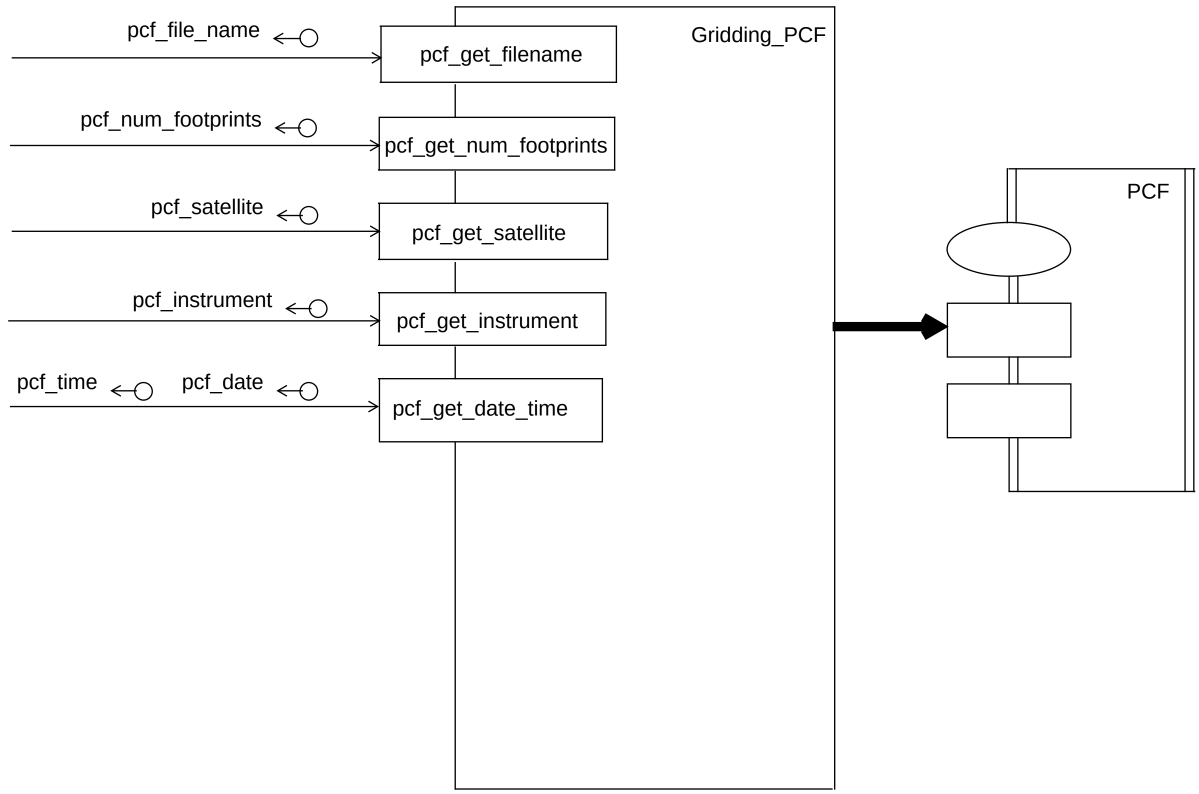


Figure 3-5. Gridding_PCF Context Diagram

3.6 Footprint

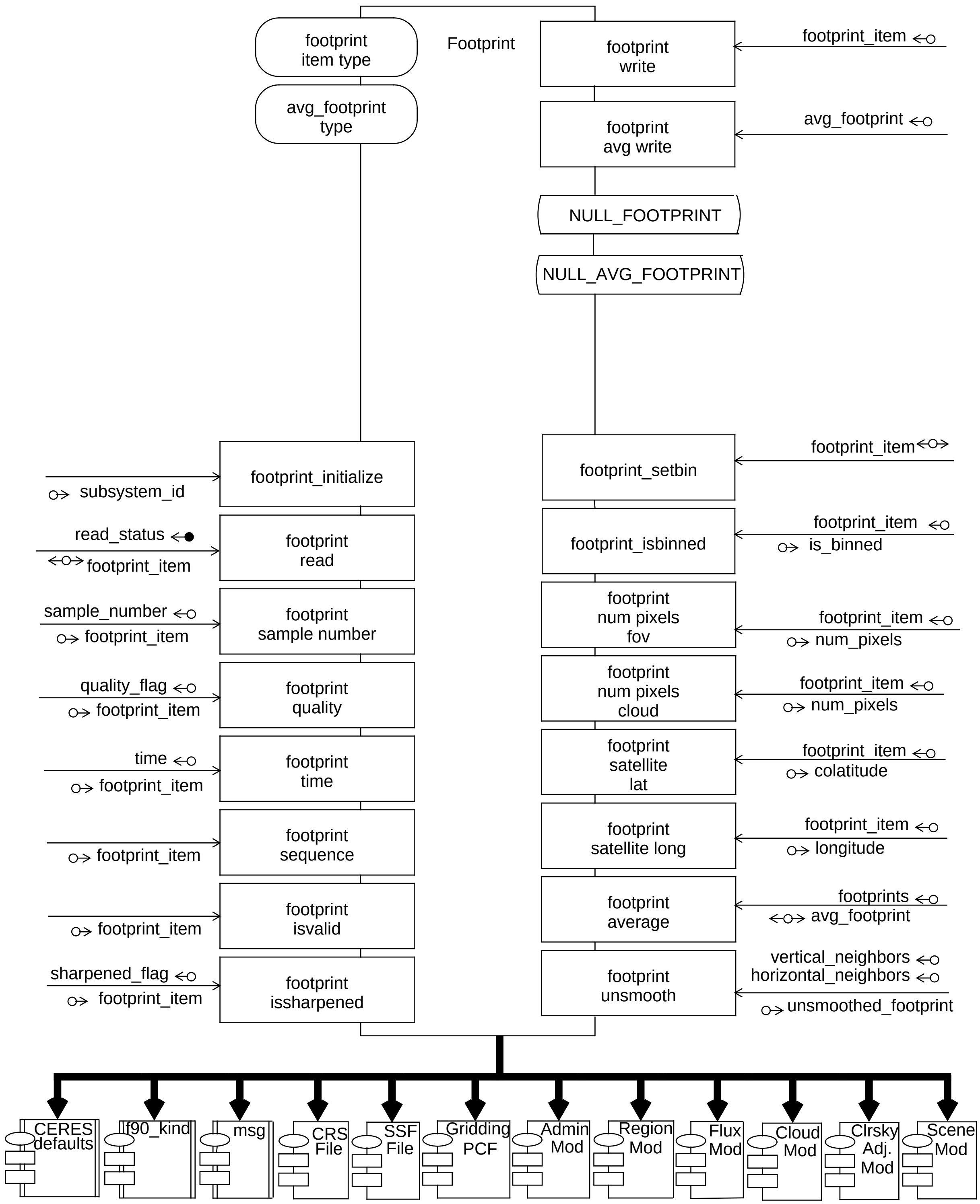


Figure 3-6. Footprint Context Diagram

3.7 Hourbox

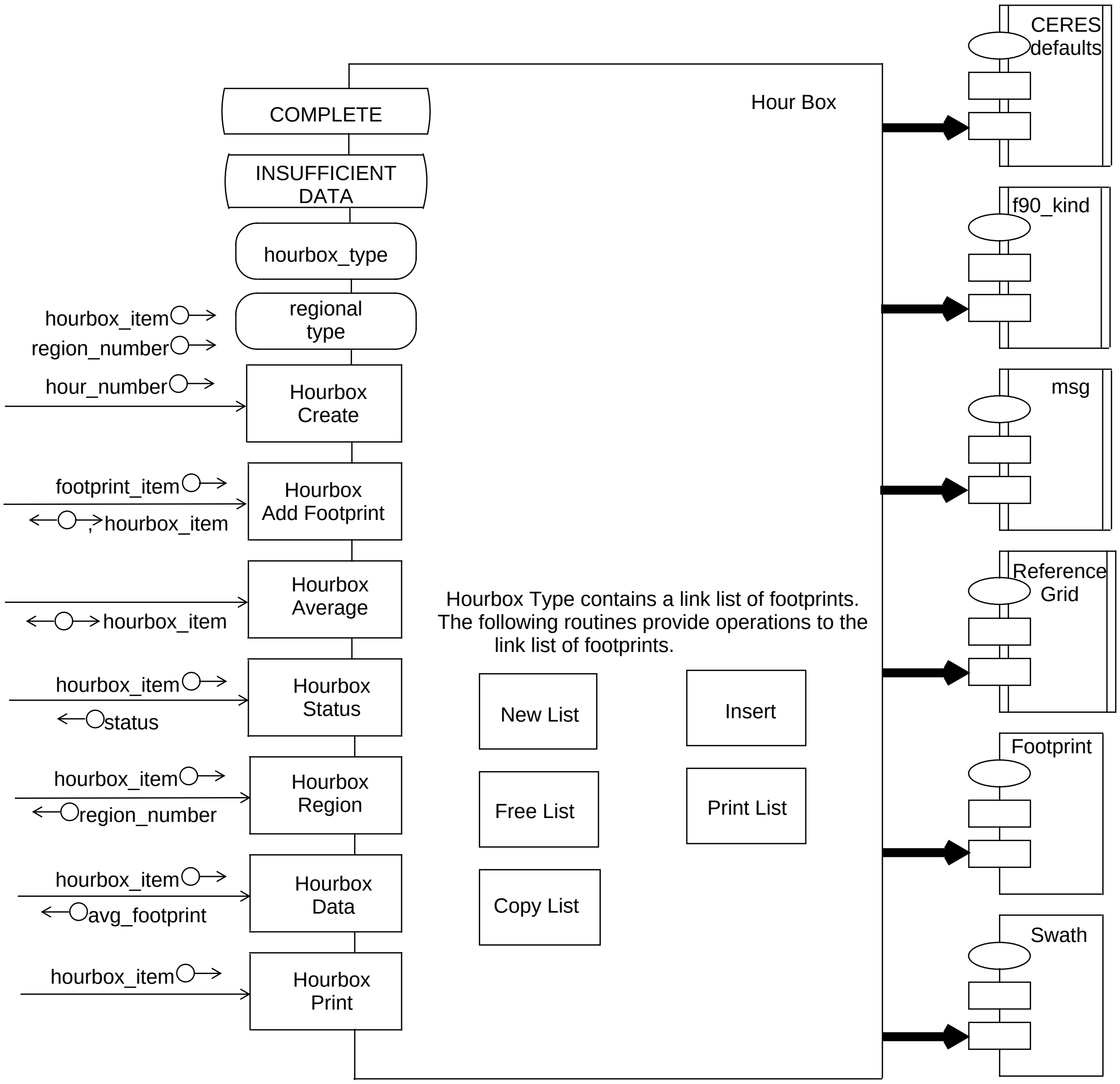


Figure 3-7. Hourbox Context Diagram

3.8 Globe

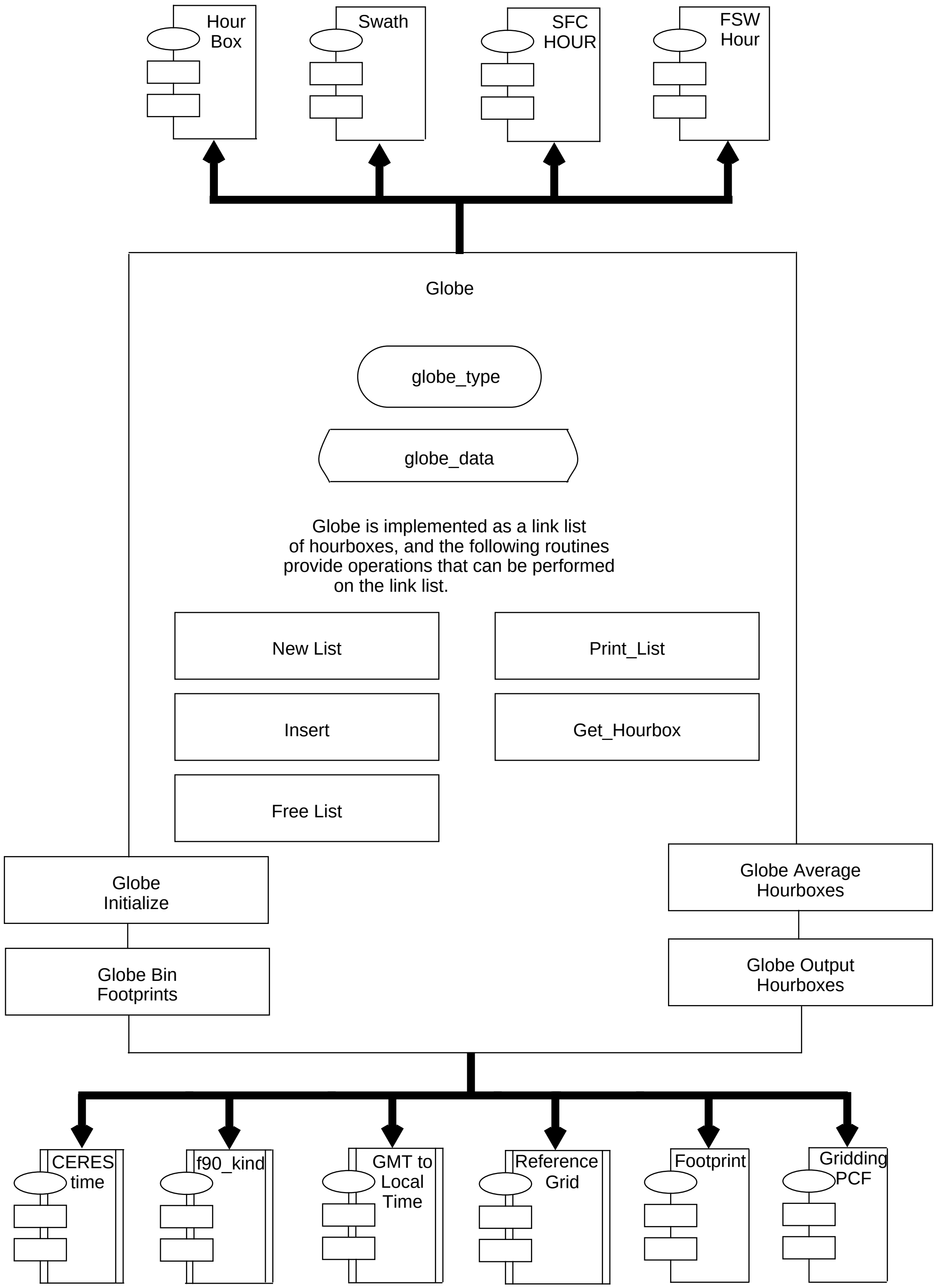


Figure 3-8. Globe Context Diagram

3.9 Swath

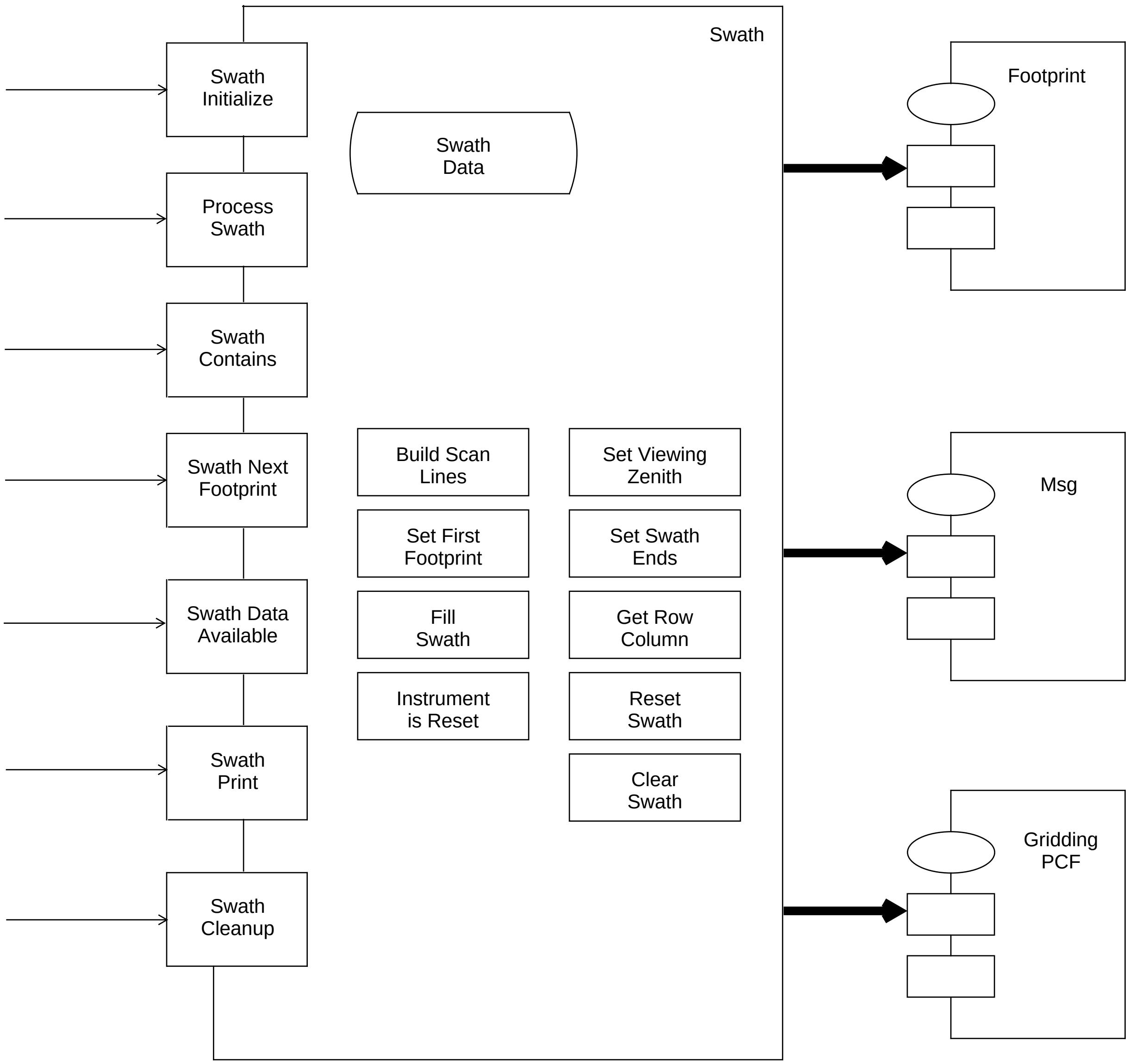


Figure 3-9. Swath Context Diagram

3.10 Cloud_Mod

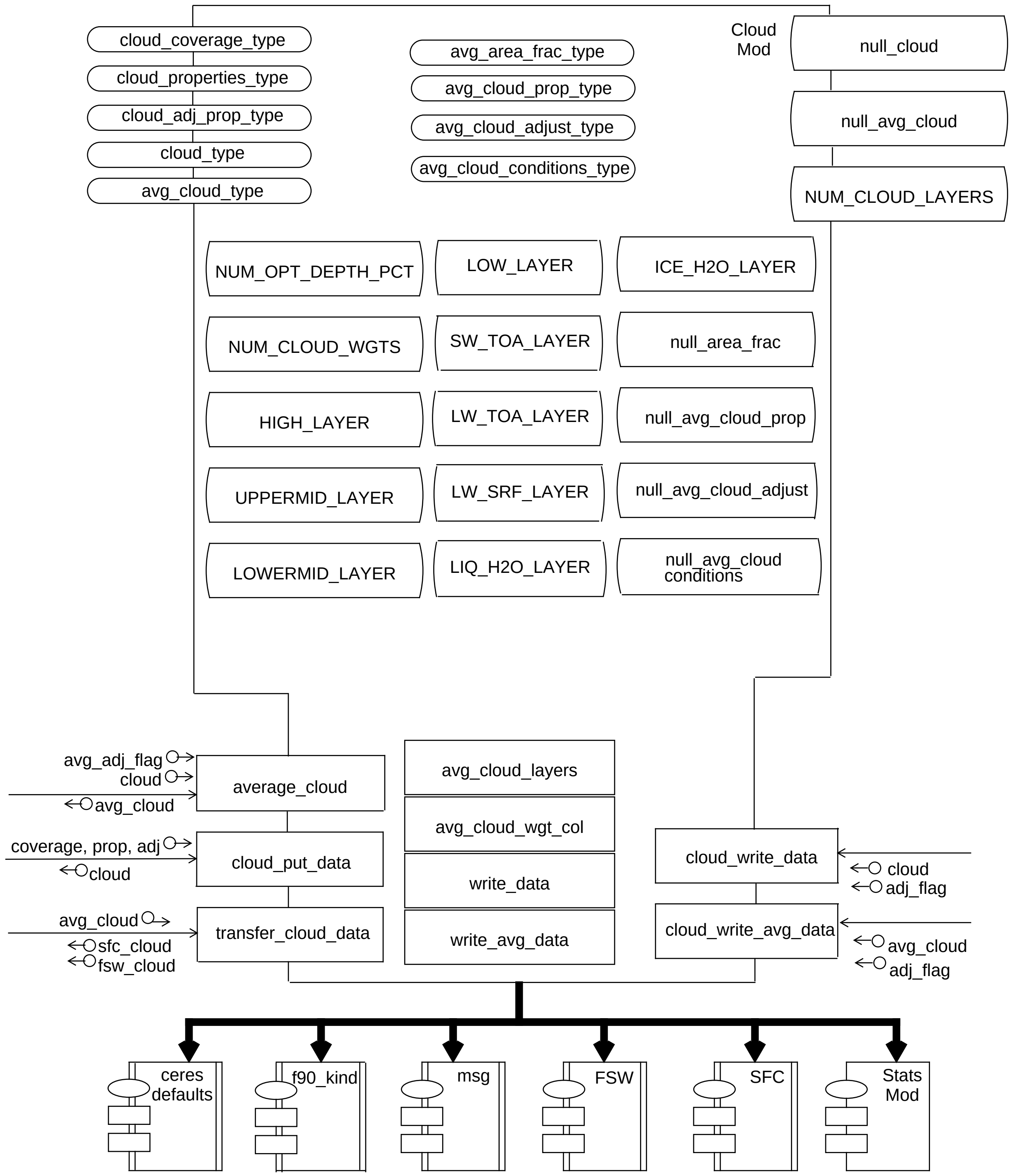


Figure 3-10. Cloud_Mod Context Diagram

3.11 Clrsky_Adj._Mod

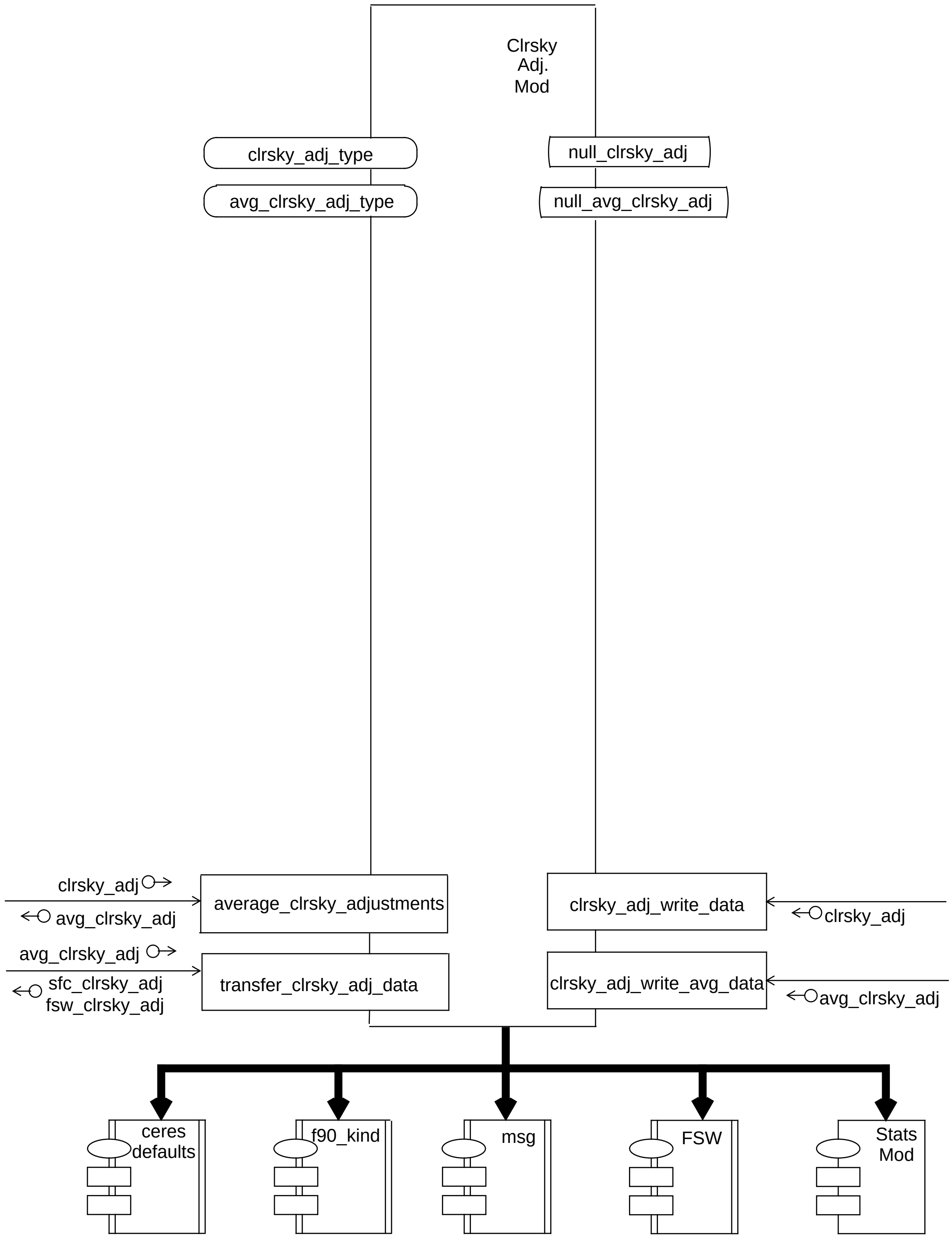


Figure 3-11. Clrsky_Adj._Mod Context Diagram

3.12 Region_Mod

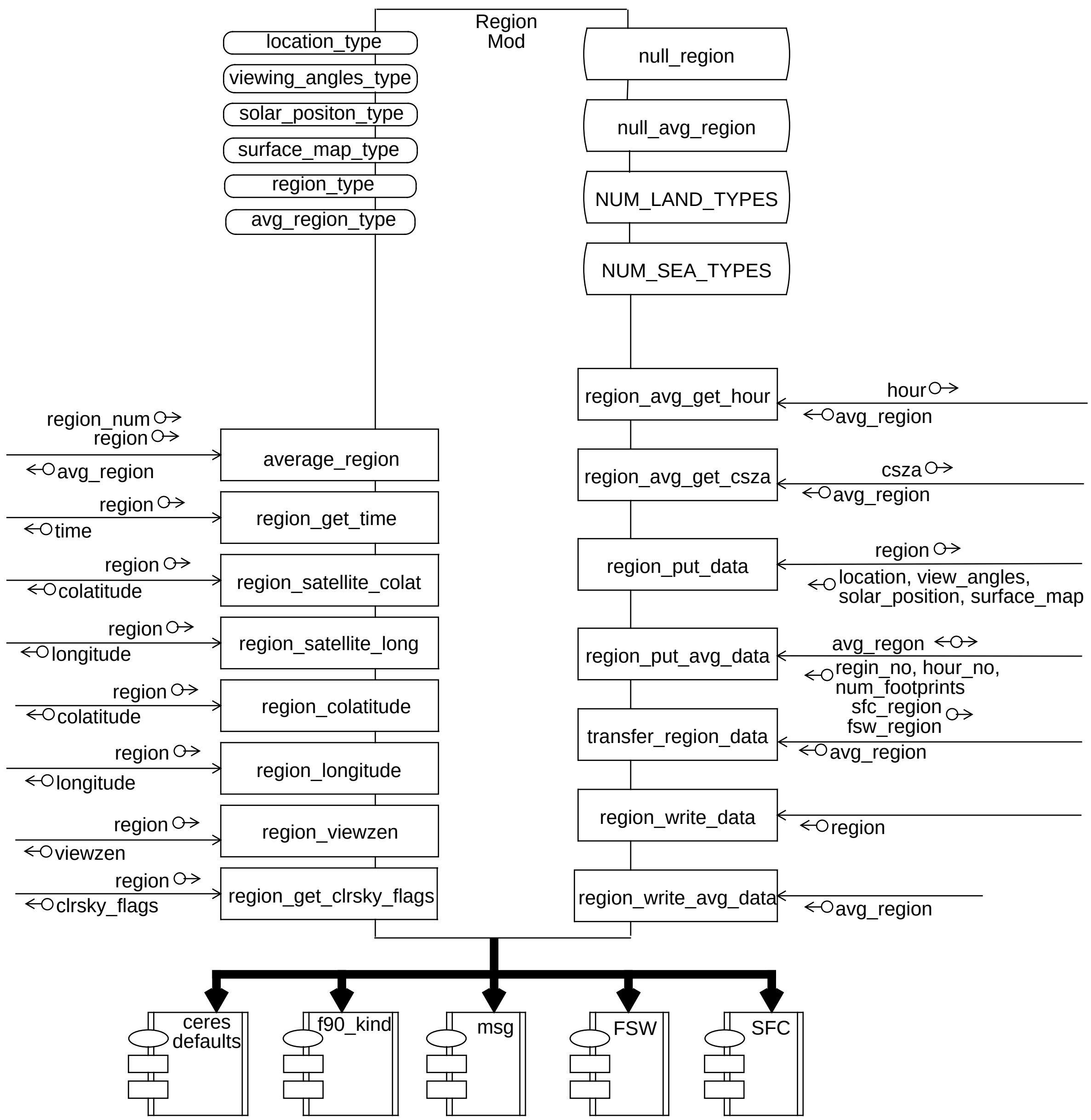


Figure 3-12. Region_Mod Context Diagram

3.13 Scene_Mod

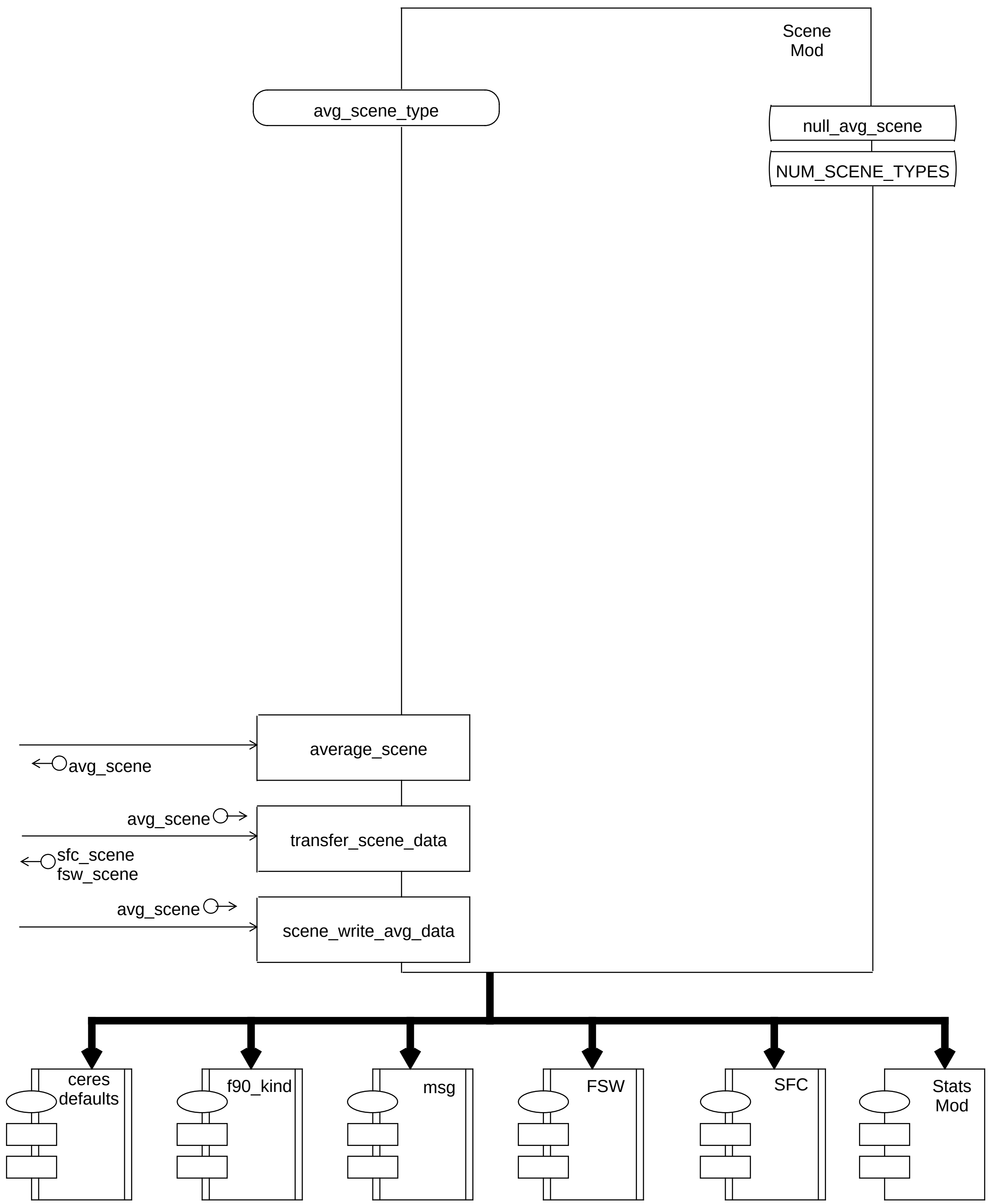


Figure 3-13. Scene_Mod Context Diagram

3.14 Stats_Mod

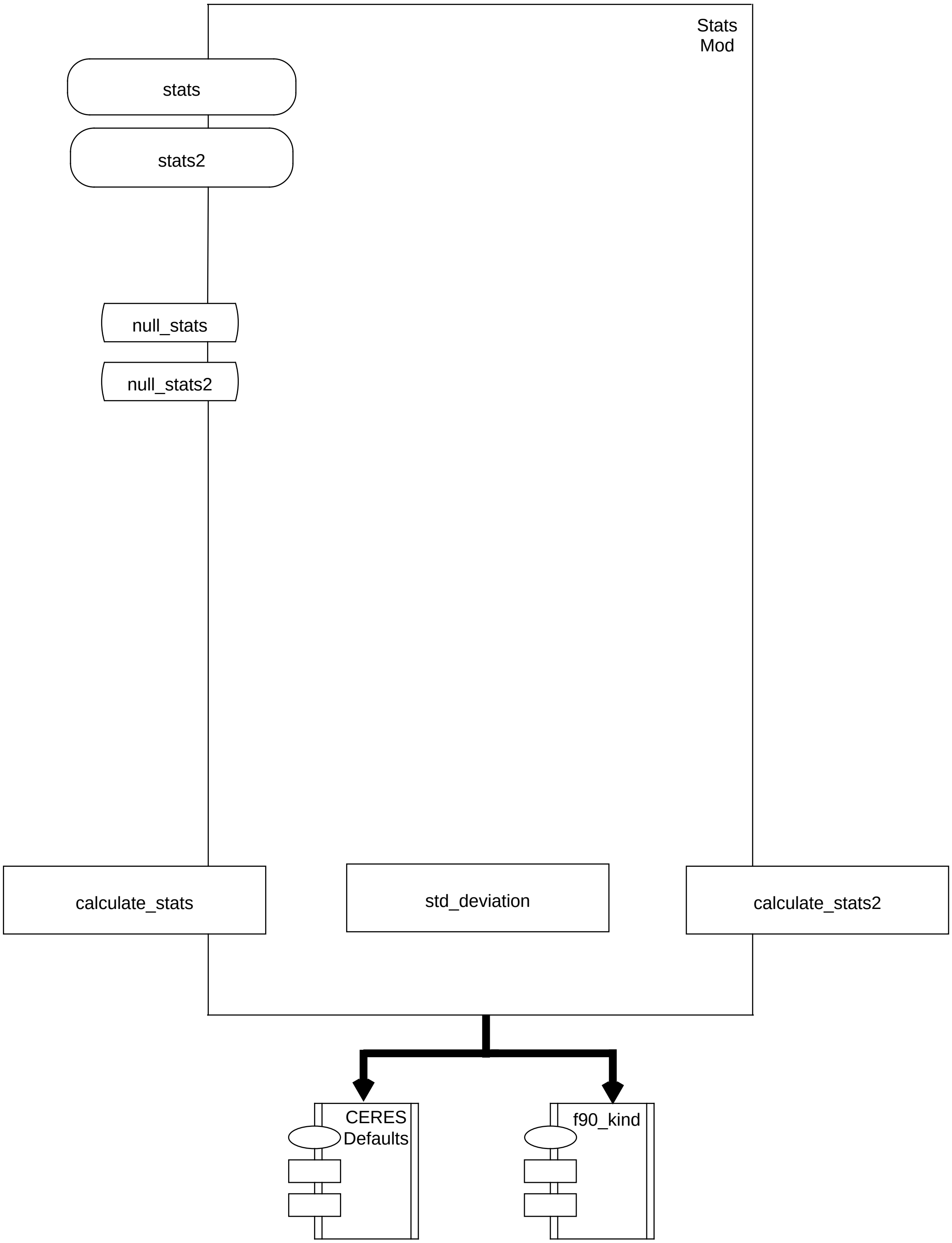


Figure 3-14. Stats_Mod Context Diagram

3.15 Post_Processor

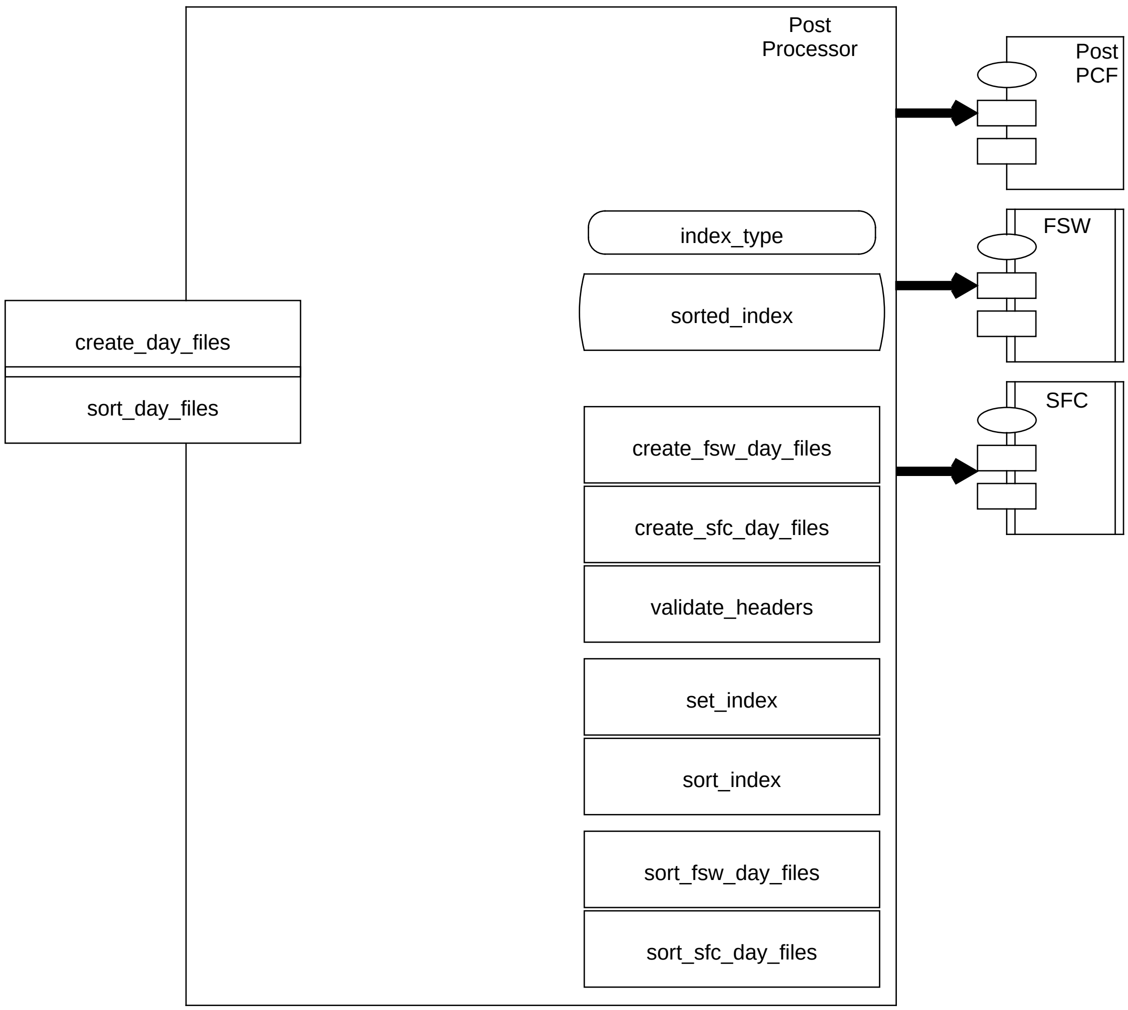


Figure 3-15. Post_Processor Context Diagram

3.16 Post_PCF

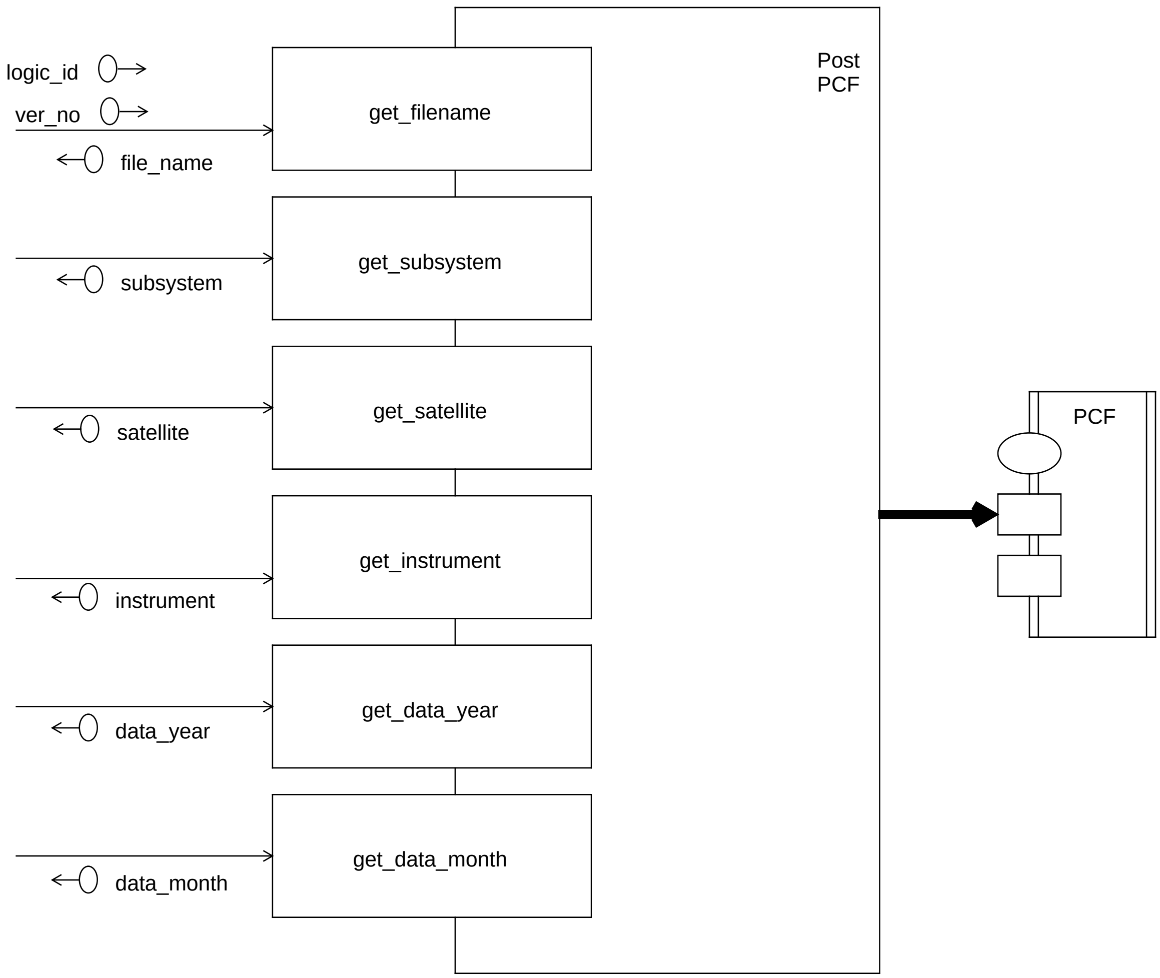


Figure 3-16. Post_PCF Context Diagram

References

1. CERES Data Management System Data Products Catalog. Release 1.0, August 1994.
2. CERES Reference Grid, CERES Software Bulletin 95-03, NASA Langley Research Center, May 1995.

Appendix A

Abbreviations and Acronyms

Appendix A
Abbreviations and Acronyms

CERES	Clouds and the Earth's Radiant Energy System
CRS	CERES Footprint Radiative Fluxes and Clouds
DAAC	Distributed Active Archive Center
EOS	Earth Observing System
EOS-AM	EOS Morning Crossing Mission
EOSDIS	EOS Data and Information System
EOS-PM	EOS Afternoon Crossing Mission
ERBE	Earth Radiation Budget Experiment
ERBS	Earth Radiation Budget Satellite
FSW	Hourly Gridded Single Satellite Fluxes and Clouds
GMT	Greenwich meridian time
NASA	National Aeronautics and Space Administration
NOAA	National Oceanic and Atmospheric Administration
PCF	Processing Control File
QC	Quality Control
SFC	Hourly Gridded Single Satellite TOA and Surface Fluxes
SSF	Single Satellite CERES Footprint TOA and Surface Fluxes
TISA	Time Interpolation and Spatial Averaging
TOA	Top-of-the-Atmosphere
TRMM	Tropical Rainfall Measuring Mission